ACEEE

Int. J. on Recent Trends in Engineering and Technology, Vol. 10, No. 1, Jan 2014

A Dynamic Botnet Detection Model based on Behavior Analysis

K.Muthumanickam[1], E.Ilavarasan[2], Sanjeev Kumar Dwivedi[3]
Research Scholar[1], Associate Professor[2], M.Tech. Student[3]
[1,2,3]Department of Computer Science and Engineering, Pondicherry Engineering College, Pondicherry, India
Email: [1]kmuthoo@pec.edu , [2]eilavarasan@pec.edu, [3]sanjeevdwivedi131988@gmail.com

***Abstract*— Today different types of malware exist in the Internet. Among them one of the malware is known as botnet which is frequently used for many cyber attacks and crimes in the Internet. Currently botnets are the main rootcause for several illegal activities like spamming, DDoS, click fraud etc. Botnets operate under the command and control(C&C) infrastructure which makes its functioning unique. As long as the Internet exists botnet also will exist. It can be used to perpetrate many Internet crimes. So fighting against them is a challenging problem. The P2P-decentralized based botnets are more dangerous than centralized botnets. In this paper a novel approach for the detection of P2P based botnet is presented. The proposed approach for the detection of botnet in the network stream analysis has been done in three phases. The first phase begins with the identification of P2P node and the second phase deals with the clustering of the suspicious P2P node. Finally botnet detection procedure has been applied which is based on stability of bots. Experimental results show that the proposed approach detects more number of bots with high accuracy.**

***Index Terms*— Bots, P2P botnet, In-out degree, Clustering, flow records, Stability.**

I. INTRODUCTION

Malicious malware can exploit vulnerabilities in the Internet computing Environment without user's knowledge. Basically malware includes viruses, worms, Trojan horses, and spyware that can be used to gather information about a computer user and access to a system without their permission. It can appear in the form of scripts, active content, code, or other software. Malware programs are divided into two classes: first class of malwares needs a host program (viruses, Trojan horses, logic bombs, trapdoors) and second class of malwares are independent programs (worms, zombie). Malwares are classified based on their characteristics; some malwares do not replicate (activated by trigger) and others that produce copies of themselves.

Botnet has become the most serious security threat on the current internet infrastructure. A botnet (BotNetwork) is an interconnected collection of compromised infected computers (bots) which is remotely controlled by its originator (called botmaster or botherder) under a common and control infrastructure [1]. Bot is a new type of malware which is designed for malicious activity. After the bot code has been installed into a computer, the computer becomes a member of the botnetwork. Here all the bots are under the control of BotMaster. So if bot exist in computer, it is not harmful until it receives command from BotMaster. After receiving the command from BotMaster, it becomes dangerous for the system. These bots are not self-

DOI: 01.IJRTET.10.1.18

propagate from one system/network to other system/network [2]. A botnet enrolls its soldiers using social engineering techniques or by exploiting software vulnerabilities.
In a decentralized architecture (P2P-based) there is no central point for communication. In this format of structure the bots can act as either client or server or both [5]. Each bot is connected to some other bots of the botnet. In this architecture botmaster can inject commands to a bot and afterwards this bot can broadcast it to all other nodes in the network. Since decentralized botnet allows commands to be injected at any node in the network, authentication of commands become essential to prevent other nodes from injecting incorrect commands. So P2P- based botnet model impose a bigger challenge for defense of network [7]. Table.1 shows different classifications of botnet based on their topological structure and some of their comparison parameters.

TABLE I. COMPARISON OF COMMAND AND CONTROL TOPOLOGIES

Topology	**Design Complexity**	**Delectability**	**Message Latency**	**Survivability**
Centralized	Low	Medium	Low	Low
Decentralized	Medium	Low	Medium	Medium
Unstructured	Low	High	High	High

This paper is presented a decentralized botnet detection model which is based on in-out degree of a node, clustering and stability of a flow. The output of one module is input for the next module. In-out degree of a node is useful to identify P2P node from a network. Because now adays attackers used decentralized architecture so it is necessary to identify P2P node. After this module, clustering algorithm is used which is based on association degree and distance between a pair of node. Finally botnet detection procedure has been used to determine bots in a network. Because all the bots have higher symmetricity in order to get command and in response. So packets related to bots are more stable than normal client packets.

The rest of the paper is organized as follows: Section II provides related works in the field of botnets. Section III describes proposed botnet detection model and section IV concludes our work.

II. RELATED WORK

A botnet is a collection of zombie computers called bots. A bot is a compromised end computer, which might be infected by malwares such as Trojan horses, backdoor, spywares and worms [16]. Bots will allow the compromised computer to be controlled by the remote attacker i.e. Botmaster in the network. The botmaster can act as head so that he/she is responsible for controlling and also issuing commands to launch many different illegal criminal activities. In 1993, Eggdrop the first Internet Relay Chat (IRC) bot was developed to automate IRC management activities [17]. These kinds of tools attract the hackers to write IRC-based malicious botnets. Thereafter, more advanced botnets such as HTTP botnets and P2P botnets [11] [18] were continued to appear.
In [3] [6], the authors discussed the impact of botnet attacks to computer world. Also they discussed the methods to create botnet, propagation and communication technique, and protocols being used. An excellent botnet tracking tool named Honeypot is used to monitor and understand the behavior of the bot operations [4].
Yukiko Sawaya et al., [8] presented a flow-based attacker detection method which detect attackers (Nuisance attackers, Simple attackers and Obvious attackers) without predefined blacklist/white list hosts. In their work, they collected and analyzed traffic flows related to the object port (open port) and decoy port (closed port). Since they used flow-based approach, attacks that were injected in payload would not be identified.
Alirena Shahrestani et al., [9] proposed visualization techniques which enhance the visibility of network traffic related to invariant bot behaviors and provide notification of existence of bots. The main advantage of this system is that visualized information is easier to be processed and for user it is easy to gain useful knowledge about bot existence in a network. Their technique will be used to visualize a limited set of invariant bot behavior (like fast response time, small size command etc.

Hsiao et al., [12] proposed a method which applied flow correlation for grouping the same activities of same bots and then scoring technique to identify normal IRC and abnormal IRC behaviors. The authors of [10] proposed a P2P botnet detection framework which is based on association between common P2P networks behaviors and host behaviors. This mechanism not only detects known P2P botnet with a high detection rate but also some unknown P2P malwares. This method deals with some problems such as data encryption, route selection, communication behaviors.

S. Zander et al., [13] Proposed P2P traffic identification based on the signature of the key packets. Their approach identified P2P flows on real-time traffic management and monitoring and based on signature of the key packets, flow attributes and extracting signature from P2P flows. This approach has Low computational complexity but generates false alarms if bots traffic is encrypted.

Yousof Al Hammadi et al., [14] proposed a behavioral correlation method for detecting P2P bots. Detection of P2P botnet has been done using Log files analysis and by correlating bot behavioral attributes. Though their method reduces false alarm, threshold to detection malicious process is unidentified.

In [15], a method to detect botnet based on flow based approach, packet sampling, payload based and pattern matching has been proposed. It is a logical chance for high speed network. But absence of payload has to be perceived.

III. PROPOSED WORK

A. System Architecture of Proposed Framework

The proposed System architecture given in Figure 1 contains the five modules, namely, packet capturing, filtering, P2P node detection, clustering, and botnet detection. In this architecture it is clear that the input of the next stage is the output of the previous stage.

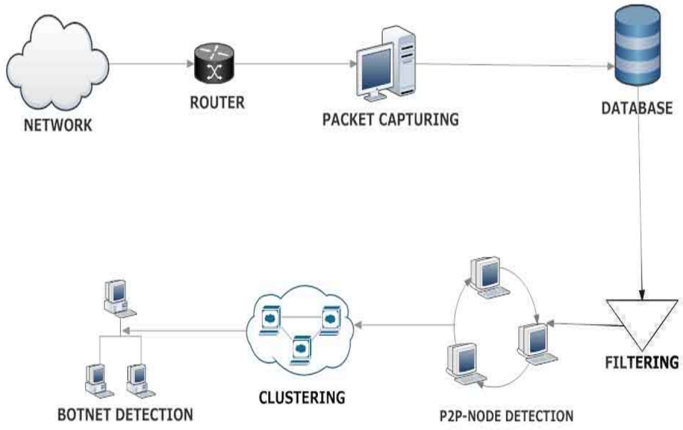

Figure1. P2P botnet detection model

The dataset about the network is analyzed such that the packets flow in the network is captured using wireshark tool in the server machine by making the network interface card as a promiscuous mode and the sniffing is carried out. This captures the flow of packet passes through the server machine. The packet captured is stored as a PCAP file in the database. The dataset may have infected host details as well as uninfected host details in the network. For capturing packets we used wireshark tool. Wireshark is a free and open source packet analyzer. Wireshark has graphical front end and plus some integrating storing and filtering option. Wireshark has rich features set e.g. deep inspection of hundreds of hundreds of protocol, multi platform, read/write many different capture file format etc.

Filtering module is responsible for filtering out unrelated traffic flows. Additionally it is responsible to filter out packets into either legitimate or suspected from the network so that only the packets, which we want to analyze are available in the network. So this stage reduces the traffic workloads.

Since goal of proposed system is to identify p2p botnet (decentralized bots) in a network, the first step is to identify p2p nodes in the network. i.e. after filter out legitimate packets, it is necessary to check how many nodes in a network received packet or message from other nodes. If any node in a network received botnet packet then according to the definition of P2P botnet, these bots broadcast command or message to the nearest nodes in the network or they find out victim nodes in the network. So at any particular time the nodes

incoming degree or outgoing degree which we call it as in-out degree is more. By using these related concepts, we implemented a new algorithm, called In-Out Degree Algorithm (IODA) which will find P2P nodes in the network. The number of P2P-nodes in the network can be identified based on the in-out degree of a node.

The next step is to cluster these P2P nodes. It is based on similar behavior or similar communication. The clustering has been done to identify the malicious peer in the network. K-Means Clustering Algorithm (KMCA) has been used for P2P-node clustering. In this algorithm the node at the center is taken as a median and then the next nearest node will be grouped and clustered as one node. The clustering can be made by means of behavior of the node. Once the behavior value of the node differs from the normal one in the cluster, then the node will be identified as suspicious node. The clustering can be made for the similar behavior of the node that has been clustered. Then the dissimilar node can be grouped as one cluster and these nodes will be analyzed in order to detect whether the node is a malicious one or not.

Finally P2P botnet traces are identified. This module uses Botnet Stability Detection Algorithm (BSDA) to detect p2p bots in a network. The input for BSDA algorithm is the output of p2p clustering algorithm. In BSDA algorithm, stability refers to the whether flows (in the context of botnet detection model flow refers to the packets which contains information about the control and command packets i.e. information about size of packets, timing information, same protocols etc.) are stable or unstable. If flows are stable it implies p2p nodes are exchanging control or command packets to other nodes and accordingly definition of p2p bots, these nodes are consider as a p2p botnet nodes. On the other hand if flows are unstable it implies p2p nodes are exchanging packets other than C&C packets and these are not a p2p botnet nodes. The detection of the botnet can be made by analyzing the control flow stability of a certain port such as UDP port. We can extract control flows from bot traffic by collecting all flows on the selected UDP port.

B. P2P Node Detection

The algorithm for P2P node detection is based on in-out degree of a node. The in-out degree of a node 's' is within the set s(i) the number of members connected in a set to the total number of members in the set which satisfies the condition.

Calculation of In-Out Degree of a Node S:

In-out degree of a node S is represented as IO(S)

Step 1: In a given specified time, collect the subset of nodes. Suppose, in time duration T, we collect subset of nodes (samples of nodes) m times.

Step 2: Now for each subset of nodes, calculate the number of connections formed by each node. In 'm' subsets, number of connection formed by node S is N(m). So total number of connections formed by a node S in all subsets = N (1) +N (2) +----------+N(m). Average number of connections formed by a node S in all m subsets (avg (S)) is N(1)+N(2)+----------+N(m) / m.

Step 3: In-out degree of a node S, in i^{th} subset is defined as the total number of connections of a node S in i^{th} subset to the average number of connection of a node S in all subsets i.e. In-out degree IO(S(i)) = N(i)/ avg (S). For each subset, we calculate the In-out degree of a node S. Let these degrees be (IOS(1),IOS(2)----------IOS(m)) corresponding to every subset.

Step 4: Maximum In-out degree of a node S is IO{S(max)}=MAX(IO{S(1)},IO{S(2)}--------IO{S(m)}).

Step 5: IO{S(maxx)} =IO{S(max)} * x%. IO{S(minn)} = IO{S(max)} * y%. Where x and y are constant**.** The value for x is 0.9 and y is 0.4.

Step 6: Now we define two sets:- Set (1) = { IO{S(i)} /IO {S(i)} > IO{S(maxx)}}. Where NOM (1) = number of members of that set. Set (2) = {IO{S(i)}/IO {S(i)} < IO{S(minn)}}. Where NOM (2) = number of members of that set.

Step 7: The In-out degree of a node S IO(S) = NOM (2) / NOM (1).

Algorithm for p2p node detection (IODA) *Input:* Set of In-out degree of a node s in all m subset = (IOS(1), IOS(2)----------IOS(m)); Initialize:- IO{S(max)},IO{S(maxx)}, IO{S(minn)} NOM (1), NOM (2).

```
Begin
For i=1 to m do
If IO{S(i)} > IO{S(maxx)}, then
NOM (1) ++;
End;
If IO{S(i)} < IO{S(minn)}, then
```

NOM (2) --;
End;
End;
IO(S) = NOM (2) / NOM (1).
If IO(S) > S (IO)
Then node S is P2P node.
End;
End;
The values of S(IO) are the statistics characteristics of IO degree of a node, which we have derived from the sample nodes.

C. *P2P Node Clustering*

For P2P node clustering, we used k-means algorithm which is based on association degree and distance between a pair of nodes. K-means clustering is a method of cluster analysis which aims to partition n observations into k clusters, in which each cluster belongs to the nearest mean.
Association degree of a pair of nodes [N(1,2)] is AD(1,2) = K(1) * [N(1,2)]+N(2,1)] + K(2) * T.
Where N(1,2) = number of packages that node 1 sent to node 2. N(2,1) = number of packages that node 2 sent to node 1. T= time of the connection between a pair of nodes (1,2) in the same period. K(1) and K(2) = constant. Harmonious degree of a pair of nodes [N(1,2)] is HD(1,2) = [N(1,2)+ N(2,1)] /ABS [N(1,2) - N(2,1)]. The distance between a pair of nodes [N(1,2)] is D(1,2) = 1 / [CD(1,2)].

The basic steps of KMCA algorithm are:

1. Specify k points as the initial centroids.
2. Repeat
3. Form k clusters by assigning all points to the closet centroid.
4. Recomputed the centroid of each cluster.
5. Until the centroids don't change.

// specified k points as the initial centroids. //. Step 1:- Select the initial-node of the C categories $(N_1)^1$, $(N_2)^1$, $(N_3)^1$ ……. $(N_C)^1$. Let K=1. // Repeat step 2, 3 and 4 till $(N_J)^{K+1} = (N_J)^K$ // Step 2:- Categorize all sampling nodes X into one specific category by K times iteration. If $D[X(N_J)^K] = \min_{(1<J<K)} D[X(N_I)^K]$ Then $X \epsilon (S_J)^K$, Where $(S_J)^K$ is the set with center node $(N_J)^K$, where=1,2,3……..C. //recomputed the centroid of each cluster. // Step 3:- Use the category $(S_J)^K$ which obtained in step2 to update the center node $(N_J)^{K+1}$ of J category, until $\sum J=1$ to C $\sum X \epsilon (S_J)^K D^2 X(N_J)^{K+1}$ to be minimum value. // until the centroids don't change. // Step 4:- With all J=1,2,3…….C, if $(N_J)^{K+1} = (N_J)^K$ The iteration ends, otherwise K = K+1, go step 2 and continue.

D. Botnet detection:

Finally botnet detection procedure which is based on stability of flow has been applied. The input for this step is the output of the previous algorithm. A sequence of packets which contain same information i.e., same source address same destination address, same source port, same destination port , same protocol etc., are called flow. The flow record contains only address information, port information, timing information, size information etc., but no information about payload. So for the detecting the stability of flows, we have to select one information from flow packets. For this we select average number of bytes per packet / flow. For simplicity, we denote this variable to avg(nbpp). For a $flow_j$ we divide the total number of bytes by the number of packets within this flow and we can obtain the value of abp of this flow. In BSDA algorithm, we use two timing windows and calculate the distribution of avg(nbpp) in different timing intervals and compare these two distributions.
The Botnet Stability Detection Algorithm (BSDA):
Step 1:-specify each value of timing window one by one.
Steps 2:- specify the value of timing interval between these two timing windows.
Steps 3:- specify the value of flow change number (initially zero) and epochs.
Steps 4:- calculate the distribution of avg(nbpp) in these two timing windows and compare.

Steps 5:- if they are closely related (difference is less than some threshold value which we calculated from previous experiments) then print “flow is stable”.
Step 6:- After first interval, the second window (second avg(nbpp)) become the baseline for second interval and again compute avg(nbpp) and compare.
Step 7;- if they are not closely related, report a flow change and increment the value of flow change number by one and repeat the same procedure.
Step 8:- Repeat this procedure till flow not change.The timing interval is same between two avg(nbpp). So by using this procedure we also able to say how many times flows changed.

IV. Conclusion

In this paper we have presented a P2P botnet detection model which is based on in-out degree of a node (IODA), clustering (KMCA) and stability of flow (BSDA). In-out degree of a node is useful to identify P2P nodes of a network. Clustering algorithm is based on harmonious degree and distance between a pair of nodes. The botnet detection model is based on stability of flow. Since all bots follow the same rule and predefined control and command method so by using this method we easily identify P2P bots from a network. This proposed botnet detection model detects both known and unknown bots.

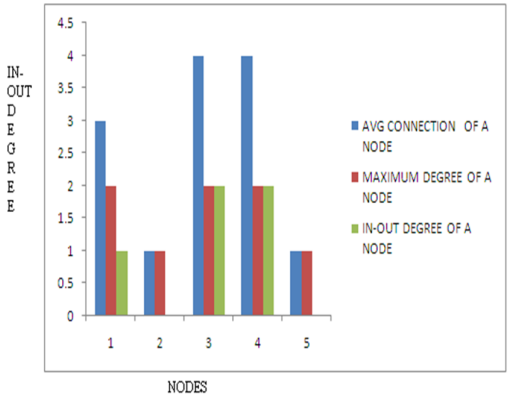

Figure 2. Relationship between avg number of connection and in-out degree of node

Figure 2 shows the relationship between average connection of nodes and in-out degree of nodes. Nodes which generate more number of connections which we refer it as ‘avg’ number of connections with other nodes having higher in-out degree than other node. i.e. they exchange more number of packets with other nodes. In above graph average connections of node (in-out degree) 3 and 4 are high than all other nodes. So this node is suspected as a P2P node while node 2 and 5 create less number of connections so these nodes are considered to be legitimate P2P node, which is purely based on in-out degree of a node.

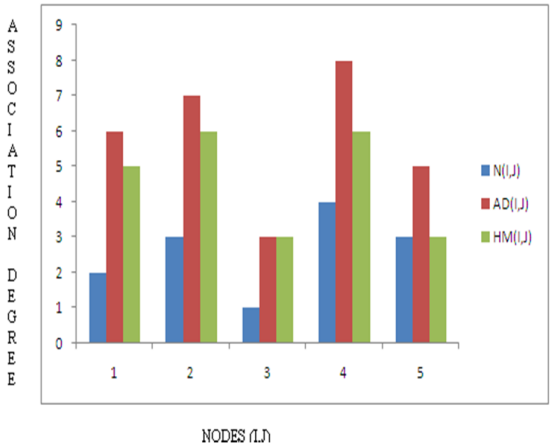

Figure 3. Relationship between association degree and harmonious degree of a pair of node

Figure 3 shows the relationship between a pair of nodes and their association degree and Harmonious degree. By observing this graph it is clear that a pair of nodes which exchanges more number of packets, their association degree and harmonious degree is higher than other nodes. This means that distance between them is less than the other pairs of nodes. As inverse relationship exists between them, it tells about its symmetricity too.

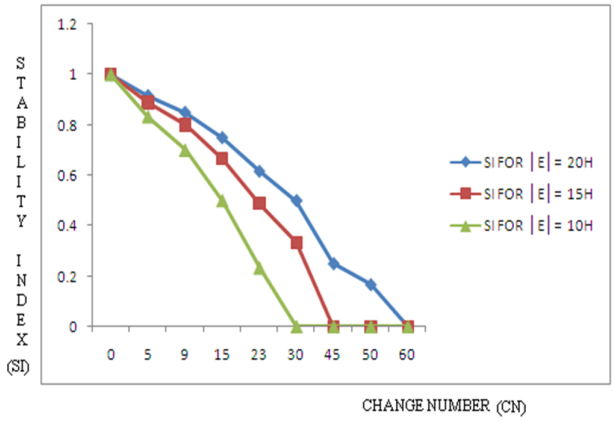

Figure 4. Stability index in different epochs

Finally Figure 4 shows how stability index vary according to control number and flow duration. As the value of change number (CN) increase in different epochs, at the same time value of stability index (SI) also increases in different epochs (keep timing interval same in all epochs). First value of CN is zero then value of SI is one in all the epochs. As the value of CN slightly increases, there is more deviation in SI.

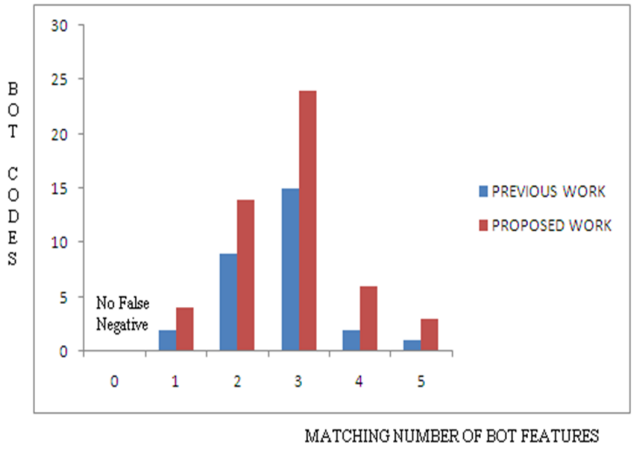

Figure 5. Matching number of bot features

The proposed work which has been implemented above using the specified standards and the previous work is taken for experiment. The experimental result of both the work is evaluated by the parameter such as matching number of bot features and stability index. The comparison graph is the evaluated results of both the work and it is clearly shows that the proposed work has better matching number of bot features and stability index compared to previous work. Figure 5 shows the matching number of bot features like attacks port, SMTP error response etc. All bots were matched with any bot features, which eliminates the incidence of any false negative on bot infected PCs. This implies that the implemented system can extract the bot process and report in a stable manner.

REFERENCES

[1] Lei Zhang, Shui Yu, Di Wu, Paul Watters, "A Survey on Latest Botnet Attack and Defense", International Joint Conference Of IEEE Trustcom-11/IEEE ICESS-11/FCST-11, pp.53-60, 2011.

[2] Maryam Feily, Alireza Shahreshtani and Sureswaran Ramadas "A Survey of Botnet and Botnet Detection",IEEE in The Third Conference of Emerging Security Information, Systems and Technologies, pp.79-84, 2009.

[3] Banday, M.T. Qadri, J.A.Shah, "Study of Botnet and Their Threats to internet Security", Sprouts:Working papers on Information System, 2009. Available at:http://sprouts.aisnet.org/9-24.

[4] Niels Provos, Thorsten Holz, "Virtual Honeypots: From Botnet Tracking to Intrusion Detection", See Chapter11- Tracking Botnets. Publisher: Addison Wesley Professional, Publishing 16 June, 2007.

[5] Hossein Rouhani Zeidanloo, Farhoud Hossswinpour and Farhood Farid Etemad "New Approach For Detection of IRC and P2P Botnet" International Journals of Computer and Electrical Engineering, Vol.2, No.6, pp.1029-1038, December, 2010.

[6] Jing Liu, Yang Xiao, Kaveh Ghaboosi, Julia Deng, Jingyuan Zhang, "Botnet: Classification, Attacks, Detection, Tracing, and Preventive Mesaures", EURASIP Journal on Wireless Communications and Networking, 2008.

[7] Mukesh Kumar, Pothula Sujatha, P. Manikandan, Madarapu, Naresh Kumar, Chetana Sidige and Sunil Kumar Verma "Self-Destructible Concentrated P2P Botnet" IJCSI International Journal of Computer Science Issues, Vol. 8, Issue 3, No.1, pp.401-405, 2011.

[8] Yukiko Sawaya, Ayumu Kubota, Yutaka Miyake "Detection of Attackers in Services Using Anomalous Host Behavior Based on Traffic Flow Statistics" PSJ International Symposium On Applications and the Internet, pp. 353-359,2011.

[9] Alireza Shahrestani, Maryam Feily Rodina Ahmad, Sureswaran Ramadass "Discovery of Invariant Bot Behavior Through Visual Network Monitoring System" Fourth International Conference on Emerging Security Information, Systems and Technologies,pp.182-188, 2010

[10] Yousof Al Hammadi, Uwe Aickelin "Behavioral Correlation for Detecting P2P Bots" Second International Conference on Future Networks, DOI 10. 1109/ICFN 2010.72, pp. 323-327, 2010.

[11] Hossein Rouhani Zeidanloo, Azizah bt Abdul Manaf "Botnet Detection by Monitoring Similar Communications Patterns" International Journals of Computer Science and Information Security, Vol. 7,No.3, pp.36-44, 2010.

[12] Hsiao-Chung Lin, Chia-Mei Chen, Jui-Yu Tzeng "Flow Based Botnet Detection" IEEE Fourth International Conference on Innovative Computing, Information and control pp.1538-1541, 2009.

[13] S. Zander, T. Nguyen and G. Armitage "P2P Traffic Identification Based on The Signature of Key Packets", IEEE conference Local Computer Networks, 2010.

[14] Yousof Al Hammadi, Uwe Aickelin "Behavioral Correlation for Detecting P2P Bots" Second International Conference on Future Networks, DOI 10. 1109/ICFN 2010.72, pp. 323-327, 2010.

[15] Mohammed Abdul Qadeer, Mohammad Zahid "Network Traffic Analysis and Intrusion Detection Using Packet Sniffer" Second International Conference on Communication Softwaret and Networks pp.313-317, 2010.

[16] Tang T.K. and Wang J.H., "The Detection of zombies, Chinese cryptography and Information Security Association (CCISA 07)". Vol.13, July 2007, pp 25-36, ISSN 17729-6056.

[17] Yang C.H. and Ting K.L.," Fast Deployment of Botnet Detection with Traffic Monitoring". Z. Fifth International Conference on Intelligent Information Hiding and Multimedia signal processing (IIHMSP 09), Sep 2009, pp 856-860.

[18] Lee J.S. Jeong, H.C., Park, J.H. Kim, M and Noh, B.N., "The Activity Analysis of Malicious HTTP-based Botnets using degree of periodic Repeatability", International conference on Security technology (SecTech 08), IEEE press, Mar 2008, pp 83-86. DOI:10.1109/SecTech.2008.52.